CHROOT-02

Call chdir("/") after using the chroot()

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-19

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4528 bytes

Attack Category	• Privilege Exploitation
Vulnerability Category	• Privilege escalation problem • Indeterminate File/Path
Software Context	
Location	
Description	Call chdir("/") after using the chroot() function. The chroot() function establishes a virtual root directory for the owning process. This may be used to limit the amount of file system access a potential hacker could use if he or she gained control of the process. Programs like ftp and httpd commonly make use of this function. One weakness of the chroot() function is that it does not work as advertised unless a chdir("/") call is issued after the chroot(). Otherwise, the current working directory could be outside the isolated hierarchy and provide the attacker with access via relative paths. Use of chroot is desirable but should also be a flag to indicate that one needs to ensure that related security issues are addressed.
APIs	**FunctionName** / **Comments** chdir / should follow soon after any chroot() call chroot / should have a chdir() call soon after
Method of Attack	An attacker who exploits another vulnerability to gain control of a program will be able to access directories other than those allowed by chroot if the programmer failed to call chdir("/").
Exception Criteria	
Solutions	**Solution Applicability** / **Solution Description** / **Solution Efficacy**

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	Whenever chroot is used.	Add a chdir("/") call ASAP following a chroot() call.	Effective at restricting filesystem access, but must ensure that other chroot issues are addressed also.

Signature Details	int chroot(const char *)

Examples of Incorrect Code	`[...]`
	`char path[] = "/usr/sandbox";`
	`chroot(path);`
	`[...]`
	`/* Continuing without changing user ID is a security risk because running as root. */`

Examples of Corrected Code	`[...]`
	`char path[] = "/usr/sandbox";`
	`close(anOpenFile); /* Should not leave file descriptors open. */`
	`if (chroot(path)) exit(1); /* Should check return value. */`
	`chdir("/"); /* Must do this or chroot() won't have intended effect */`
	`setegid(ogid); /* Should change group ID */`
	`seteuid(ouid); /* Should change user ID */`
	`[...]`
	`/* Now can safely continue */`

Source Reference	• Viega, John & McGraw, Gary. Building Secure Software: How to Avoid Security Problems the Right Way. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, pg. 205.
Recommended Resources	• chroot man page[2] • Bishop, Matt & Dilger, Michael. Checking for Race Conditions in File Accesses[3], 1996
Discriminant Set	**Operating System**: • UNIX (All) **Languages**: • C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com